```
function [f_vec] = program(a0,b0,mu0,lambda0,mu_vec,obs,N,threshold,with_prior)

    diff = 1000;
    y_= (1/N)*sum(obs);
    x = 0;
    y = 100;

    %inizialize a,b,m and s squared with prior
    if( with_prior == true)
        a_start = a0;
        b_start = b0;
        m_start = mu0;
        s_squ_start = b0/(a0*lambda0);
    else
        a_start = (y-x) *rand;
        b_start = (y-x) *rand;
        m_start = (y-x) *rand;
        s_squ_start = (y-x) *rand;
    end


    %calculate free energy
    f_vec = [];
    F_start = -a_start* log(b_start)+ gammaln(a_start)-gammaln(a0)+a0*log(b0)+0.5*log(lambda0)+log(sqrt(s_squ_start))-N/2*log(2*pi)+0.5;
    i =1;

    %update parameters and calculate new free energy, stop if threshold is
    %reached
    while diff >= threshold
        s_squ = 1/(a_start/b_start*(N+lambda0));
        m = (lambda0*mu0+N*y_)/(lambda0+N);
        a = a0 + (N+1)/2;
        b = b0 + 0.5 *((obs-mu_vec)'*(obs-mu_vec)+ lambda0*(mu0-m_start)^2+(N+lambda0)*s_squ_start);
        F = -a* log(b)+ gammaln(a)-gammaln(a0)+a0*log(b0)+0.5*log(lambda0)+log(sqrt(s_squ))-N/2*log(2*pi)+0.5;
        diff = abs(F_start-F);


        %set start values to calculated values
        f_vec= [f_vec, F_start];
        a_start = a;
        b_start=  b;
        m_start = m;
        s_squ_start = s_squ;
        F_start = F;
        i = i+1;
    end

end
```

Published with MATLAB® R2019a